Amendments to the Specification:

1.) Please replace the paragraph beginning at page 3, line 16, with the following rewritten paragraph:

FIGURE 1 illustrates a conventional successive interference cancellation (SIC) receiver unit for three codes (k) (i.e., three user signals) and two stages (110, 130); ~~(i.e., interference canceled twice).~~ *i.e.*, interference canceled twice. Symbols for a data stream are detected and used to cancel their influence on the received symbols of other data streams before detecting the next data stream symbols. User data streams' influence are preferably canceled in order of their descending signal strength, since the strongest data stream contributes the most to the multiple access interference (MAI) in the system. Thus, the influence of the strongest signal received for a data stream is canceled from all other data streams. Another reason for canceling the strongest signal first is that the data estimate in that signal usually is the most reliable one. The interference cancellation process can be repeated several times, e.g., two stages as shown in FIGURE 1, to improve the reliability of the data estimation.

2.) Please replace the paragraph beginning at page 4, line 12, with the following rewritten paragraph:

Considering the first stage of the SIC scheme, the data sequence r is received 102 by the receiver and supplied to an Interference Cancellation Unit (ICU) 110a associated with the first spreading code, shown within dashed lines. The data sequence 102, received by the ICU is despread 112a by a spreading code convolved with the channel estimate of user channel (hereinafter referred to as convolution sequence and denoted S_K), and is preferably normalized such that $S_k^* S_k=1$, where S_k^* is the complex conjugate of S_k. As shown in Figure 1, the received data sequence is despread 112a by S_k^* ~~(114a)~~, RAKE receiver function, and added 120a to the previously detected soft values (previously stored RAKE receiver result) 122a to form the estimate symbol of the user 124a. However, in the first stage there is no previous detected soft values, and the soft values of the despread sequence (x_{11}) are the same as the soft values of the despread sequence (y_{11}) after the addition operation 120a. The despread sequence (y_{11}) is stored to be used in later stages. Meanwhile, the soft values (result of RAKE receiver function 112a) are respread 116a to enable subtraction 126a from the received data sequence r (102) used for the second code. After subtracting the respread soft values from the received data sequence, the result is used in a similar process as described above for the second user code using the ICU 110b. Specifically, the output respread signal of the second ICU 110b is subtracted 126b from the data sequence r1 to produce data sequence r2 having the first and second user estimate influence removed therefrom. A further similar process is used for a third user using ICU 110c. After all users' symbol estimates have been removed from the received signal, the remainder (r3) is employed to perform the whole interference cancellation operation described above, a determined number of stages thereafter. As shown in Figure 1, a second stage 130 is employed which performs the same operations as the previous stage. The second stage is mainly used to correct the inaccuracy of each user's symbol estimate, which is due to the non-orthogonality between the codes. The non-orthogonality between the codes prevents the precise contribution of a user's signal to be extracted exactly while ignoring the contribution of other users signals. However, the contribution detected in the second stage is added to the contribution extracted in the first stage and used to provide a better estimation. Processing delays 128a, 128b and 128c are introduced in the system to delay the data sequence before the subtraction operations 126a, 126b and 126c for each code interference cancellation operation and at each stage (iteration). These delays account for the time used in the despreading and the respreading operations in the ICU. The despread data sequences within the ICUs for each user and at each stage are soft values, and there is no hard decision (discrete values) or other mapping function applied inside the ICUs. It should be understood that despite the fact that the ICUs do not perform a hard decision, after the last stage, a hard decision should be performed to transform the last accumulated estimate into a symbol decision. This can be performed by a hard-decision device (not shown in the figure).

3.) Please replace the paragraph beginning at page 7, line 5, with the following rewritten paragraph:

FIGURE 2A illustrates a similar receiver implementing a parallel interference cancellation (PIC) scheme. The PIC scheme is similar to the SIC scheme with minor differences. The received data sequence r 202 is supplied to all the ICUs (210a-c) corresponding to all the users' at a specific stage, which reduces the processing delay in the receiver compared to the SIC scheme. The received data sequence r 202 is despread (212a-c) and respread (216a-c), as explained above, by all user's codes before the influence/contribution of all the user's codes signals are subtracted (226a-c) concurrently from the received data sequence r 202. In other words, all the ICUs, at each stage, despread the same data sequence before their contribution is subtracted. Consequently, the processing delay 228 introduced to compensate for the delay in the despreading and respreading operations is reduced to only one delay for all the users at each stage compared to multiple delays in the SIC scheme.

4.) Please replace the paragraph beginning at page 15, line 13, with the following rewritten paragraph:

The received data sequence symbol is first loaded into register 1 (302), and then switch 1 (306) transfers the loaded symbol to register 2 (304) at the specified time instances. The symbol in register 2 (304) is despread (334) by the complex conjugate (314) of the convolution sequence S_k corresponding to the data sequence of the user. The result of the despreading operation, using a user spreading code, is optionally saved in an accumulator associated with the user's spreading code. The function of saving the despread sequences is performed by a third switch 318 which switches between accumulators 320a-K for storing/accumulating the despread sequence in select registers. Each of the registers/accumulators corresponds to a different spreading code. In other words, a data sequence associated with a spreading code is only stored in the accumulator associated with that same spreading code. Each accumulation operation (340a,...,340K) is performed to add the contribution of a despread data sequence, by a user's spreading code, captured at all stages for a received data sequence. This accumulation is similar to the addition operation performed in several stages in conventional ICUs. For example, the addition operations 120a and 140a and any other addition operation in different stages thereafter for the same user spreading code, shown in Figure 1, can be performed by adder 340a, using output register 1 (320a) as a storage/accumulator unit.

5.) Please replace the paragraph beginning at page 16, line 18, with the following rewritten paragraph:

In the meantime, the despread sequence of the despreader 334 is respread 336 by the convolution sequence S_k. The output respread sequence is subtracted <u>(330)</u> from a time delayed version <u>(332)</u> of the original sequence and the result loaded into register 1. The above described process is then repeated for each spreading code and for each stage using the same hardware elements.

6.) Please replace the paragraph beginning at page 18, line 14, with the following rewritten paragraph:

FIGURE 5 illustrates an exemplary timing diagram for the operation of the receiver of FIGURE 3. The number of users in this example is four, i.e., K=4, and the number of iterations is set equal to two. The variable n is set equal to 2, which makes the receiver (300) a HIC scheme receiver. Hence, users in this receiver are detected in groups of two. As mentioned previously, the registers are first cleared (502) and the received data sequence is loaded into register 1 (504). The first switch is then closed momentarily (506) to move the information in register 1 to register 2. Switch 2 is then moved to position 1 (512), which is the convolution sequence associated with the spreading code of the first user. The despreading is then performed (514) for the first user in the ICU. Switch 3 is then moved to the first location (516) to accumulate the despread signal of user 1 in the output register 1 (518). In the meantime, the despread signal of user 1 is respread (520) and subtracted from the information in register 1 (522). The result of this subtraction is then used to update register 1 (522). Switch 2 is then moved to position 2 ~~(512)~~ (532), which is the convolution sequence associated with the spreading code of the second user. The despreading is then performed (534), in the ICU, on the data sequence associated with the second spreading code. Switch 3 is then moved to the second location (536) to accumulate the despread signal of spreading code 2 in the output register 2 (538). In the meantime, the despread signal of user 2 is respread (540) and subtracted from the information in register 1, and register 1 is updated (542). Switch 1 is then closed to move information in register 1 to register 2 (544). The previously described process (steps 512-540) is then repeated for spreading codes 3 and 4 ~~(550 and 570)~~ (process 550, steps 552-560 and process 590, steps 572-580), followed by an update of register 1 in steps 562 to 582, respectively). Switch 1 is then closed to move the information in register 1 to register 2 (step 584). The data sequence is now despread for all spreading codes in the first iteration/stage (590) and the whole process is repeated a second time (591) to account for the inaccuracy in the first stage. At the end of the two iterations/stages, the accumulated output is propagated (594) to other components in the system. The process is then repeated for other received data sequences for all the spreading codes. It is clear that the example provided above, is a HIC scheme detecting the first two spreading code's signals in parallel and canceling their contribution concurrently. The signals of the next two spreading codes, 3 and 4, are also detected in parallel, and their contribution is canceled concurrently with each other. However, the group of spreading codes (group 1 having spreading codes 1 and 2, and group 2 having spreading codes 3 and 4) are detected and canceled successively.

7.) Please replace the paragraph beginning at page 22, line 18, with the following rewritten paragraph:

FIGURE 6 illustrates an ICU (600) in which the same elements are used for both the respreading and despreading operations according to a preferred embodiment of the present invention. The ICU (600) utilizes registers and switches, as shown in Figure 6, to perform the despreading and respreading operations. The ICU (600) first receives the data sequence 602 which is loaded into register 1 (610). In the meantime, the complex conjugate (606) spreading code (604) corresponding to the desired user is loaded into register 2 (612). Register 1 shifts its contents out to be multiplied by the output-shifted contents of register 2. This shifting process is performed on a chip-wise basis to allow chip-wise multiplication of the contents using multiplier 614. The shifting operation is performed until the contents of the registers are all multiplied (the multiplication is performed a number of times equal to the length of the registers). The product of each chip-wise multiplication is added (624) to an accumulator after each multiplication. During this operation the switch 620 is moved to position 1. After the completion of the chip-wise multiplication operation and the accumulation of the multiplication product, the content of the accumulator, the despread data sequence, is fed to the output accumulator 630. At the same time as the accumulated value in 622 is fed to the output accumulator 630, the same accumulator content is loaded to all positions of register 1 (610). The respreading operation is then performed using the same hardware elements used in the above-mentioned despreading operation. A chip-wise multiplication 614 is then performed by multiplying the sequence in register 1 (610) with S_k (626) contained in register 2 (612) to generate the respread symbol 632. The multiplication is a chip-wise multiplication that operates by shifting the contents of both registers out and multiplying the shifted output. The product of the multiplication is fed back by moving the switch 620 to position 2 to enable subtraction 330 from the received sequence. The operation is repeated for different user codes and for different stages as described above.